

# Modal Inclusion Logic: Being Lax is Simpler than Being Strict

Lauri Hella<sup>1</sup>, Antti Kuusisto<sup>2</sup>, Arne Meier<sup>3</sup>, and Heribert Vollmer<sup>3</sup>

<sup>1</sup> School of Information Sciences, University of Tampere, Kanslerinrinne 1 B,  
33014 University of Tampere, Finland, lauri.hella@uta.fi

<sup>2</sup> Dept. of Philosophy, Stockholm University, SE-106 91 Stockholm, Sweden, and  
DTU Compute, Technical University of Denmark, Richard Petersens Plads 324,  
DK-2800, Kgs. Lyngby, Denmark. antti.j.kuusisto@gmail.com

<sup>3</sup> Institut für Theoretische Informatik, Leibniz Universität Hannover, Appelstr. 4,  
30167 Hannover, Germany, {meier,vollmer}@thi.uni-hannover.de

**Abstract.** We investigate the computational complexity of the satisfiability problem of modal inclusion logic. We distinguish two variants of the problem: one for strict and another one for lax semantics. The complexity of the lax version turns out to be complete for EXPTIME, whereas with strict semantics, the problem becomes NEXPTIME-complete.

## 1 Introduction

Dependence logic was introduced by Jouko Väänänen [12] in 2007. It is a first-order logic that enables one to explicitly talk about dependencies between variables. It thereby generalizes Henkin quantifiers and also, in a sense, Hintikka's independence-friendly logic. Dependence logic can be used to formalize phenomena from a plethora of scientific disciplines such as database theory, social choice theory, cryptography, quantum physics, and others. It extends first-order logic by specific terms  $\text{dep}(x_1, \dots, x_{n-1}, x_n)$  known as dependence atoms, expressing that the value of the variable  $x_n$  depends on the values of  $x_1, \dots, x_{n-1}$ , i.e.,  $x_n$  is functionally determined by  $x_1, \dots, x_{n-1}$ . As such a dependence does not make sense when talking about single assignments, formulas are evaluated over sets of assignments, called *teams*. The semantics of the atom  $\text{dep}(x_1, \dots, x_{n-1}, x_n)$  is defined such that it is true in a team  $T$  if in the set of all assignments in  $T$ , the value of  $x_n$  is functionally determined by the values of  $x_1, \dots, x_{n-1}$ .

In addition to dependence atoms, also generalized dependency atoms have been introduced in the literature. Examples include the independence atom (asserting that two sets of variables are informationally independent in a team), the non-emptiness atom (asserting that the team is non-empty), and, most importantly to the present paper, the inclusion atom  $\vec{x} \subseteq \vec{y}$  for vectors of variables  $\vec{x}, \vec{y}$ , asserting that in a team, the set of tuples assigned to  $\vec{x}$  is included in the set of tuples assigned to  $\vec{y}$ . This corresponds to the definition of inclusion dependencies in database theory, which state that all tuples of values taken by the attributes  $\vec{x}$  are also taken by the attributes  $\vec{y}$ .

Väänänen [13] also introduced dependence atoms into modal logic. There teams are sets of worlds, and a dependence atom  $\text{dep}(p_1, \dots, p_{n-1}, p_n)$  holds in a team  $T$  if there is a Boolean function that determines the value of  $p_n$  from the values of  $p_1, \dots, p_{n-1}$  in each world in  $T$ . The so obtained modal dependence logic MDL was studied from the point of view of expressivity and complexity in [11]. Following the above mentioned developments in first-order dependence logic, modal dependence logic was also extended by generalized dependency atoms in [6], such as, e.g., independence atoms and inclusion atoms.

In the context of first-order dependence logic and its variants, two alternative kinds of team semantics have been distinguished, *lax* and *strict semantics* [2]. Lax semantics is the standard team semantics, while for strict semantics, some additional uniqueness or strictness properties are required. In the modal context, this mainly concerns the diamond modality  $\Diamond$ . Usually, i.e., in lax semantics, a formula  $\Diamond\varphi$  holds in a team  $T$  if there is a team  $S$  such that every world in  $T$  has at least one successor in  $S$  and  $\varphi$  holds in  $S$ . (Also, the worlds in  $S$  are required to have a predecessor in  $T$ .) In strict semantics, we require that  $S$  contains, for every world in  $T$ , a unique successor given by a surjection  $f : T \rightarrow S$ . (In first-order logic, strict semantics for the existential quantifier is defined similarly.) In both the modal and the first-order context, the operator known as *splitjunction* is also defined differently for lax and strict semantics (see Section 2 below).

For many variants of first-order and modal dependence logic, there is no distinction in expressive power between the two semantics. However, the choice of semantics plays a role in independence and inclusion logics, i.e., team semantics over (first-order) logics with the independence and inclusion atoms. For example, in the first-order case, inclusion logic under strict semantics has the same expressive power as dependence logic, i.e., ESO (existential second order logic) [3] and hence NP, while under lax semantics it is equivalent to greatest fixpoint logic and hence can express exactly the polynomial-time decidable properties over finite ordered structures.

The purpose of the present paper is to exhibit a further context in which a quite dramatic difference between the two flavours of team semantics exists. We turn to modal inclusion logic and study the computational complexity of its satisfiability problem. For lax semantics, we show EXPTIME-completeness by proving the upper bound via a translation to a variant of PDL, and the lower bound by a reduction from a succinct encoding of a P-complete problem. Satisfiability under strict semantics is shown NEXPTIME-complete using a translation into two-variable logic with counting (upper bound) and a chain of reductions from a dependence version of QBF-validity (lower bound). The complexity difference also holds for the finite satisfiability problem.

## 2 Preliminaries

Let  $\Pi$  be a countably infinite set of proposition symbols. The set of formulas of *modal inclusion logic*  $\text{MInc}$  is defined inductively by the following grammar.

$$\varphi ::= p \mid \neg p \mid (\varphi_1 \wedge \varphi_2) \mid (\varphi_1 \vee \varphi_2) \mid p_1 \cdots p_k \subseteq q_1 \cdots q_k \mid \Box\varphi \mid \Diamond\varphi,$$

where  $p, p_1, \dots, p_k, q_1, \dots, q_k \in \Pi$  are proposition symbols and  $k$  is any positive integer. The formulas  $p_1 \cdots p_k \subseteq q_1 \cdots q_k$  are called *inclusion atoms*. For a set  $\Phi \subseteq \Pi$ , we let  $\text{MInc}(\Phi)$  be the sublanguage where propositions from  $\Phi$  are used. Observe that formulas are essentially in negation normal form; negations may occur only in front of proposition symbols.

A Kripke model is a structure  $M = (W, R, V)$ , where  $W \neq \emptyset$  is a set (the domain of the model, or the set of worlds/states),  $R \subseteq W \times W$  is a binary relation (the accessibility or transition relation), and  $V: \Pi \rightarrow \mathcal{P}(W)$  is a *valuation* interpreting the proposition symbols. Here  $\mathcal{P}$  denotes the power set operator.

The language of basic unimodal logic is the sublanguage of  $\text{MInc}$  without formulas  $p_1 \cdots p_k \subseteq q_1 \cdots q_k$ . We assume that the reader is familiar with standard Kripke semantics of modal logic; we let  $M, w \Vdash \varphi$  denote the assertion that the point  $w \in W$  of the model  $M$  satisfies  $\varphi$  according to standard Kripke semantics. We use the symbol  $\Vdash$  in order to refer to satisfaction according to standard Kripke semantics, while the symbol  $\models$  will be reserved for *team semantics*, to be defined below, which is the semantics  $\text{MInc}$  is based on.

Let  $T$  be a subset of the domain  $W$  of a Kripke model  $M$ . The set  $T$  is called a *team*. The semantics of the inclusion atoms  $p_1 \cdots p_k \subseteq q_1 \cdots q_k$  is defined such that  $M, T \models p_1 \cdots p_k \subseteq q_1 \cdots q_k$  iff for each  $u \in T$ , there exists a point  $v \in T$  such that  $\bigwedge_{i \in \{1, \dots, k\}} (u \in V(p_i) \Leftrightarrow v \in V(q_i))$ . The intuition here is that every vector of truth values taken by  $p_1, \dots, p_k$ , is included in the set of vectors of truth values taken by  $q_1, \dots, q_k$ .

Let  $M = (W, R, V)$  be a Kripke model and  $T \subseteq W$  a team. Define the set of successors of  $T \subseteq W$  to be  $R(T) := \{s \in W \mid \exists s' \in T : (s', s) \in R\}$ . Also define  $R\langle T \rangle := \{T' \subseteq W \mid \forall s \in T \exists s' \in T' \text{ s.t. } (s, s') \in R \text{ and } \forall s' \in T' \exists s \in T \text{ s.t. } (s, s') \in R\}$ , the set of legal successor teams. The following clauses together with the above clause for inclusion atoms define *lax semantics* for  $\text{MInc}$ .

$$\begin{aligned}
M, T \models^\ell p &\Leftrightarrow w \in V(p) \text{ for all } w \in T. \\
M, T \models^\ell \neg p &\Leftrightarrow w \notin V(p) \text{ holds for all } w \in T. \\
M, T \models^\ell \varphi \wedge \psi &\Leftrightarrow M, T \models^\ell \varphi \text{ and } M, T \models^\ell \psi. \\
M, T \models^\ell \varphi \vee \psi &\Leftrightarrow M, S \models^\ell \varphi \text{ and } M, S' \models^\ell \psi \text{ for some } S, S' \subseteq T \text{ such that} \\
&\quad \text{we have } S \cup S' = T. \\
M, T \models^\ell \Box \varphi &\Leftrightarrow M, R(T) \models^\ell \varphi. \\
M, T \models^\ell \Diamond \varphi &\Leftrightarrow \exists T' \in R\langle T \rangle : M, T' \models^\ell \varphi
\end{aligned}$$

The other semantics for  $\text{MInc}$ , *strict semantics*, differs from the lax semantics only in its treatment of the disjunction  $\vee$  and diamond  $\Diamond$ . Therefore, all other clauses in the the definition of  $\models^s$  are the same as those for  $\models^\ell$ . The clauses for  $\vee$  and  $\Diamond$  in strict semantics are as follows.

$$\begin{aligned}
M, T \models^s \varphi \vee \psi &\Leftrightarrow M, S \models^s \varphi \text{ and } M, S' \models^s \psi \text{ for some } S, S' \subseteq T \text{ such that} \\
&\quad S \cup S' = T \text{ and } S \cap S' = \emptyset. \\
M, T \models^s \Diamond \varphi &\Leftrightarrow M, f(T) \models^s \varphi \text{ for some function } f: T \rightarrow W \text{ such that} \\
&\quad (u, f(u)) \in R \text{ for all } u \in T. \text{ (Here } f(T) = \{f(u) \mid u \in T\}.)
\end{aligned}$$

The difference between lax and strict semantics is as the terms suggest. In strict semantics, the division of a team with the splitjunction  $\vee$  is strict; no point is allowed to occur in both parts of the division contrarily to lax semantics. For  $\Diamond$ , strictness is related to the use of functions when finding a team of successors.

It is well known and easy to show that for a formula  $\varphi$  of modal logic, i.e., a formula of **MInc** *without* inclusion atoms,  $M, T \models^\ell \varphi$  iff  $\forall w \in T (M, w \Vdash \varphi)$ , where  $\Vdash$  denotes satisfaction in the standard sense of Kripke semantics. The same equivalence holds for  $\models^s$ . This is the so-called *flatness* property.

The satisfiability problem of **MInc** with lax (strict) semantics, is the problem that asks, given a formula  $\varphi$  of **MInc**, whether there exists a nonempty team  $T$  and a model such that  $M, T \models^\ell \varphi$  ( $M, T \models^s \varphi$ ) holds. Two different problems arise, depending on whether lax or strict semantics is used. The corresponding finite satisfiability problems require that a satisfying model has a finite domain.

### 3 Computational Complexity

#### 3.1 Upper bound for lax semantics

In this section we show that the satisfiability and finite satisfiability problems of **MInc** with lax semantics are in EXPTIME. The result is established by an equivalence preserving translation to *propositional dynamic logic* extended with the global and converse modalities. It is well-known that this logic is complete for EXPTIME (see [1, 5, 14]). In fact, we will only need multimodal logic with the global modality and converse modalities for our purposes.

Let  $\Pi$  and  $\mathcal{R}$  be countably infinite sets of proposition and binary relation symbols, respectively. We define the following modal language  $\mathcal{L}$  via  $\varphi ::= p \mid \neg\varphi \mid (\varphi_1 \wedge \varphi_2) \mid \langle R \rangle\varphi \mid \langle R^{-1} \rangle\varphi \mid \langle E \rangle\varphi$ . Here  $p \in \Pi$ ,  $R \in \mathcal{R}$ , and  $E$  is a novel symbol. The (classical Kripke-style) semantics of  $\mathcal{L}$  is defined with respect to ordinary pointed Kripke models  $(M, w)$  for multimodal logic. Let  $M = (W, \{R\}_{R \in \mathcal{R}}, V)$  be a Kripke model, where  $V: \Pi \rightarrow \mathcal{P}(W)$  is the *valuation* function interpreting proposition symbols. The following clauses define the semantics of  $\mathcal{L}$  (notice that we use the turnstile  $\Vdash$  instead of  $\models$ , which is reserved for team semantics in this paper).

$$\begin{aligned} M, w \Vdash p &\Leftrightarrow w \in V(p) & \text{and} & & M, w \Vdash \neg\varphi &\Leftrightarrow M, w \not\Vdash \varphi \\ M, w \Vdash \varphi_1 \wedge \varphi_2 &\Leftrightarrow M, w \Vdash \varphi_1 \text{ and } M, w \Vdash \varphi_2 \\ M, w \Vdash \langle R \rangle\varphi &\Leftrightarrow M, u \Vdash \varphi \text{ for some } u \text{ such that } wRu \\ M, w \Vdash \langle R^{-1} \rangle\varphi &\Leftrightarrow M, u \Vdash \varphi \text{ for some } u \text{ such that } uRw \\ M, w \Vdash \langle E \rangle\varphi &\Leftrightarrow M, u \Vdash \varphi \text{ for some } u \in W \end{aligned}$$

We next define a satisfiability preserving translation from modal inclusion logic to  $\mathcal{L}$ . We let  $[R]$  and  $[E]$  denote  $\neg\langle R \rangle\neg$  and  $\neg\langle E \rangle\neg$ , respectively. Before we fix the translation, we define some auxiliary formulas.

Let  $\theta$  be a formula of **MInc**. We let  $SUB(\theta)$  denote the set of subformulas of  $\theta$ ; we distinguish all instances of subformulas, so for example  $p \wedge p$  has *three* subformulas (the right and the left instances of  $p$  and the conjunction itself).

For each formula  $\varphi \in SUB(\theta)$ , fix a fresh proposition symbol  $p_\varphi$  that does not occur in  $\theta$ . We next define, for each  $\varphi \in SUB(\theta)$ , a novel auxiliary formula  $\chi_\varphi$ .

If  $\varphi \in SUB(\theta)$  is a literal  $p$  or  $\neg p$ , we define  $\chi_\varphi := [E](p_\varphi \rightarrow \varphi)$ .

Now fix a symbol  $R \in \mathcal{R}$ , which will ultimately correspond to the diamond used in modal inclusion logic. For the remaining subformulas  $\varphi$  of  $\theta$ , with the exception of inclusion atoms, the formula  $\chi_\varphi$  is defined as follows.

1.  $\chi_{\varphi \wedge \psi} := [E]( (p_{\varphi \wedge \psi} \leftrightarrow p_\varphi) \wedge (p_{\varphi \wedge \psi} \leftrightarrow p_\psi) )$
2.  $\chi_{\varphi \vee \psi} := [E]( p_{\varphi \vee \psi} \leftrightarrow (p_\varphi \vee p_\psi) )$
3.  $\chi_{\Box \varphi} := [E]( (p_{\Box \varphi} \rightarrow [R]p_\varphi) \wedge (p_\varphi \rightarrow \langle R^{-1} \rangle p_{\Box \varphi}) )$
4.  $\chi_{\Diamond \varphi} := [E]( (p_{\Diamond \varphi} \rightarrow \langle R \rangle p_\varphi) \wedge (p_\varphi \rightarrow \langle R^{-1} \rangle p_{\Diamond \varphi}) )$

We then define the formulas  $\chi_\alpha$ , where  $\alpha \in SUB(\theta)$  is an inclusion atom. We appoint a fresh binary relation  $R_\alpha$  for each inclusion atom in  $\theta$ . Assume  $\alpha$  denotes the inclusion atom  $p_1 \cdots p_k \subseteq q_1 \cdots q_k$ . We define

$$\begin{aligned} \chi_\alpha^+ &:= \bigwedge_{i \in \{1, \dots, k\}} [E]( (p_\alpha \wedge p_i) \rightarrow \langle R_\alpha \rangle (p_\alpha \wedge q_i) ), \\ \chi_\alpha^- &:= \bigwedge_{i \in \{1, \dots, k\}} [E]( (p_\alpha \wedge \neg p_i) \rightarrow \langle R_\alpha \rangle (p_\alpha \wedge \neg q_i) ), \\ \chi_\alpha &:= \chi_\alpha^+ \wedge \chi_\alpha^- \wedge \bigwedge_{i \in \{1, \dots, k\}} [E]( \langle R_\alpha \rangle q_i \rightarrow [R_\alpha] q_i ). \end{aligned}$$

Finally, we define  $\varphi_\theta := p_\theta \wedge \bigwedge_{\varphi \in SUB(\theta)} \chi_\varphi$ .

**Theorem 1.** *The satisfiability and finite satisfiability problems for modal inclusion logic with lax semantics are in EXPTIME.*

*Proof.* We will show that any formula  $\theta$  of modal inclusion logic is satisfiable iff its translation  $\varphi_\theta$  is. Furthermore,  $\theta$  is satisfiable over a domain  $W$  iff  $\varphi_\theta$  is satisfiable over  $W$ , whence we also get the desired result for finite satisfiability;  $\mathcal{L}$  has the finite model property since it clearly translates to two-variable logic via a simple extension of the *standard translation* (see [1] for the standard translation).

Let  $M = (W, R, V)$  be a Kripke model. Let  $I(\theta) \subseteq SUB(\theta)$  be the set of inclusion atoms in  $\theta$ . Assume that  $M, X \models^\ell \theta$ , where  $X$  is a nonempty team. We next define a multimodal Kripke model  $N := (W, R, \{R_\alpha\}_{\alpha \in I(\theta)}, V \cup U)$ , where  $U: \{p_\varphi \mid \varphi \in SUB(\theta)\} \rightarrow \mathcal{P}(W)$  extends the valuation function  $V$ .

Define  $U(p_\theta) = X$ . Thus we have  $M, U(p_\theta) \models^\ell \theta$ . Working from the root towards the leaves of the parse tree of  $\theta$ , we next interpret the remaining predicates  $p_\varphi$  inductively such that the condition  $M, U(p_\varphi) \models^\ell \varphi$  is maintained.

Assume  $U(p_{\psi \wedge \psi'})$  has been defined. We define  $U(p_\psi) = U(p_{\psi'}) = U(p_{\psi \wedge \psi'})$ . As  $M, U(p_{\psi \wedge \psi'}) \models^\ell \psi \wedge \psi'$ , we have  $M, U(p_\psi) \models^\ell \psi$  and  $M, U(p_{\psi'}) \models^\ell \psi'$ . Assume then that  $U(p_{\psi \vee \psi'})$  has been defined. Thus there exist sets  $S$  and  $S'$  such that  $M, S \models^\ell \psi$  and  $M, S' \models^\ell \psi'$ , and furthermore,  $S \cup S' = U(p_{\psi \vee \psi'})$ . We define  $U(p_\psi) = S$  and  $U(p_{\psi'}) = S'$ . Consider then the case where  $U(p_{\Diamond \varphi})$  has been defined. Call  $T := U(p_{\Diamond \varphi})$ . As  $M, T \models^\ell \Diamond \varphi$ , there exists a set  $T' \subseteq W$  such that each point in  $T$  has an  $R$ -successor in  $T'$ , and each point in  $T'$  has an

$R$ -predecessor in  $T$ , and furthermore,  $M, T' \models^\ell \varphi$ . We set  $U(p_\varphi) := T'$ . Finally, in the case for  $p_{\Box\varphi}$ , the set  $U(p_\varphi)$  is defined to be the set of points that have an  $R$ -predecessor in  $U(p_{\Box\varphi})$ .

We have now fixed an interpretation for each of the predicates  $p_\varphi$ . The relations  $R_\alpha$ , where  $\alpha$  is an inclusion atom, remain to be interpreted. Let  $p_1 \cdots p_k \subseteq q_1 \cdots q_k$  be an inclusion atom in  $\theta$ , and denote this atom by  $\alpha$ . Call  $T := U(p_\alpha)$ . Let  $u \in T$ . Since  $M, T \models^\ell \alpha$ , there exists a point  $v \in T$  such that for each  $i \in \{1, \dots, k\}$ ,  $u \in V(p_i)$  iff  $v \in V(q_i)$ . Define the pair  $(u, v)$  to be in  $R_\alpha$ . In this fashion, consider each point  $u$  in  $T$  and find exactly one corresponding point  $v$  for  $u$ , and put the pair  $(u, v)$  into  $R_\alpha$ . This fixes the interpretation of  $R_\alpha$ .

Let  $w \in X = U(p_\theta)$ . Recalling how the sets  $U(p_\varphi)$  were defined, it is now routine to check that  $N, w \models \varphi_\theta$ .

We then consider the converse implication of the current theorem. Thus we assume that  $N, w \models \varphi_\theta$ , where  $N$  is some multimodal Kripke model in the signature of  $\varphi_\theta$  and  $w$  a point in the domain of  $N$ . We let  $W$  denote the domain and  $V$  the valuation function of  $N$ .

For each  $\varphi \in SUB(\theta)$ , define the team  $X_\varphi := V(p_\varphi)$ . We will show by induction on the structure of  $\theta$  that for each  $\varphi \in SUB(\theta)$ , we have  $N, X_\varphi \models^\ell \varphi$ . Once this is done, it is clear that  $M, X_\theta \models^\ell \theta$ , where  $M$  is the restriction of  $N$  to the signature of  $\theta$ , and we have  $X_\theta \neq \emptyset$ .

Now recall the definition of the formulas  $\chi_\varphi$ , where  $\varphi \in SUB(\theta)$ . Let  $p \in SUB(\theta)$ . It is clear that  $N, X_p \models^\ell p$ , since  $N, w \models \chi_p$ . Similarly, we infer that  $N, X_{\neg q} \models^\ell \neg q$  for  $\neg q \in SUB(\theta)$ .

Consider then a subformula  $p_1 \cdots p_k \subseteq q_1 \cdots q_k$  of  $\varphi$ . Denote this inclusion atom by  $\alpha$ . Consider a point  $u \in X_\alpha$ . If  $u$  satisfies  $p_i$  for some  $i \in \{1, \dots, k\}$ , then we infer that since  $N, w \models \chi_\alpha^+$ , there exists a point  $v_i \in X_\alpha$  that satisfies  $q_i$ . Similarly, if  $u$  satisfies  $\neg p_j$ , we infer that since  $N, w \models \chi_\alpha^-$ , there exists a point  $v_j \in X_\alpha$  that satisfies  $\neg q_j$ . To conclude that  $N, X_\alpha \models^\ell \alpha$ , it suffices to show that all such points  $v_i$  and  $v_j$  can be chosen such that  $v_i = v_j$  for all  $i, j \in \{1, \dots, k\}$ . This follows due to the third conjunct of  $\chi_\alpha$ .

Having established the basis of the induction, the rest of the argument is straightforward. We consider explicitly only the case where the subformula under consideration is  $\Diamond\varphi$ . Here we simply need to argue that for each  $u \in X_{\Diamond\varphi}$ , there exists a point  $v \in X_\varphi$  such that  $uRv$ , and for each  $u' \in X_\varphi$ , there exists a point  $v' \in X_{\Diamond\varphi}$  such that  $v'Ru'$ . This follows directly, since  $N, w \models \chi_{\Diamond\varphi}$ .  $\square$

### 3.2 Lower bound for lax semantics

In this section we prove that the satisfiability problem of **MInc** with lax semantics, **MInc-lax-SAT**, is hard for **EXPTIME**. We do this by reducing the succinct version of the following P-hard problem to it which is closely related to the problem **PATH SYSTEMS** [4, p. 171].

**Definition 1.** Let **PER** be the following problem: An instance of **PER** is a structure  $\mathfrak{A} = (A, S)$  with  $A = \{1, \dots, n\}$  and  $S \subseteq A^3$ . A subset  $P$  of  $A$  is  $S$ -persistent

if it satisfies the condition  $(*)$  if  $i \in P$ , then there are  $j, k \in P$  such that  $(i, j, k) \in S$ .  $\mathfrak{A}$  is a positive instance if  $n \in P$  for some  $S$ -persistent set  $P \subseteq A$ .

It is well known that structures  $(A, S)$  as above can be represented in a succinct form by using Boolean circuits. Namely if  $C$  is Boolean circuit with  $3 \cdot l$  input gates then it defines a structure  $\mathfrak{A}_C = (A_C, S_C)$  given below. We use here the notation  $\sharp(a_1, \dots, a_l)$  for the natural number  $i$ , whose binary representation is  $(a_1, \dots, a_l)$ . Let  $A_C = \{1, \dots, 2^l\}$ , and for all  $i, j, k \in A$ , let  $(i, j, k) \in S_C$  if and only if  $C$  accepts the input tuple  $(a_1, \dots, a_l, b_1, \dots, b_l, c_1, \dots, c_l) \in \{0, 1\}^{3l}$ , where  $i = \sharp(a_1, \dots, a_l)$ ,  $j = \sharp(b_1, \dots, b_l)$  and  $k = \sharp(c_1, \dots, c_l)$ . We say that  $C$  is a succinct representation of  $\mathfrak{A}_C$ .

**Definition 2.** The succinct version of PER, S-PER, is the following problem: An instance of S-PER is a circuit  $C$  with  $3l$  input gates.  $C$  is a positive instance, if  $\mathfrak{A}_C$  is a positive instance of PER.

**Proposition 1.** S-PER is EXPTIME-hard with respect to PSPACE reductions.

*Proof.* (Idea.) The succinct version of the CIRCUIT VALUE problem is polynomial space reducible to S-PER. Since succinct CIRCUIT VALUE is known to be EXPTIME-complete (see [8, Section 20]), the claim follows. For the details of the proof, see the appendix.  $\square$

We will next show that S-PER is polynomial time reducible to the satisfiability problem of MInc with lax semantics, and hence the latter is also EXPTIME-hard. In the proof we use the following notation: If  $T$  is a team and  $p_1, \dots, p_n$  are proposition symbols, then  $T(p_1, \dots, p_n)$  is the set of all tuples  $(a_1, \dots, a_n) \in \{0, 1\}^n$  such that for some  $w \in T$ ,  $a_t = 1 \iff w \in V(p_t)$  for  $t \in \{1, \dots, n\}$ . Note that the semantics of inclusion atoms can now be expressed as

$$M, T \models p_1 \cdots p_n \subseteq q_1 \cdots q_n \iff T(p_1, \dots, p_n) \subseteq T(q_1, \dots, q_n).$$

**Theorem 2.** The satisfiability and finite satisfiability problems for MInc with lax semantics are hard for EXPTIME with respect to PSPACE reductions.

*Proof.* Let  $C$  be a Boolean circuit with  $3l$  input gates. Let  $g_1, \dots, g_m$  be the gates of  $C$ , where  $g_1, \dots, g_{3l}$  are the input gates and  $g_m$  is the output gate. We fix a distinct Boolean variable  $p_i$  for each gate  $g_i$ . Let  $\Phi$  be the set  $\{p_1, \dots, p_m\}$  of proposition symbols. We define for each  $i \in \{3l + 1, \dots, m\}$  a formula  $\theta_i \in \text{MInc}(\Phi)$  that describes the correct operation of the gate  $g_i$ :

$$\theta_i = \begin{cases} p_i \leftrightarrow \neg p_j & \text{if } g_i \text{ is a NOT gate with input } g_j \\ p_i \leftrightarrow (p_j \wedge p_k) & \text{if } g_i \text{ is an AND gate with inputs } g_j \text{ and } g_k \\ p_i \leftrightarrow (p_j \vee p_k) & \text{if } g_i \text{ is an OR gate with inputs } g_j \text{ and } g_k \end{cases}$$

Let  $\psi_C$  be the formula  $(\bigwedge_{3l+1 \leq i \leq m} \theta_i) \wedge p_m$ . Thus,  $\psi_C$  essentially says that the truth values of  $p_i$ ,  $1 \leq i \leq m$ , match an accepting computation of  $C$ .

Now we can define a formula  $\varphi_C$  of  $\text{MInc}(\Phi)$  which is satisfiable if and only if  $C$  is a positive instance of S-PER. For the sake of readability, we denote here the variables corresponding to the input gates  $g_{l+1}, \dots, g_{2l}$  by  $q_1, \dots, q_l$ . Similarly, we denote the variables  $p_{2l+1}, \dots, p_{3l}$  by  $r_1, \dots, r_l$ .

$$\varphi_C := \psi_C \wedge q_1 \cdots q_l \subseteq p_1 \cdots p_l \wedge r_1 \cdots r_l \subseteq p_1 \cdots p_l \wedge p_m \cdots p_m \subseteq p_1 \cdots p_l.$$

Note that  $\varphi_C$  can clearly be constructed from the circuit  $C$  in polynomial time.

Assume first that  $\varphi_C$  is satisfiable. Thus there is a Kripke model  $M = (W, R, V)$  and a nonempty team  $T$  of  $M$  such that  $M, T \models^\ell \varphi_C$ . Consider the model  $\mathfrak{A}_C = (A_C, S_C)$  that corresponds to the circuit  $C$ . We define a subset  $P$  of  $A_C$  as follows:  $P := \{\#(a_1, \dots, a_l) \mid (a_1, \dots, a_l) \in T(p_1, \dots, p_l)\}$ .

Observe first that since  $M, T \models^\ell p_m$  and  $M, T \models^\ell p_m \cdots p_m \subseteq p_1 \cdots p_l$ ,  $(1, \dots, 1) \in T(p_1, \dots, p_l)$  and hence  $2^l = \#(1, \dots, 1) \in P$ . Thus, it suffices to show that  $P$  is  $S_C$ -persistent. To prove this, assume that  $i = \#(a_1, \dots, a_l) \in P$ . Then there is a state  $w \in T$  such that  $w \in V(p_t) \iff a_t = 1$  for  $1 \leq t \leq l$ .

Define now  $b_t, c_t \in \{0, 1\}$ ,  $1 \leq t \leq l$ , by the condition

$$b_t = 1 \iff w \in V(q_t) \quad \text{and} \quad c_t = 1 \iff w \in V(r_t).$$

As  $M, T \models^\ell \psi_C$ , it follows from flatness that  $M, w \models \psi_C$ . By the definition of  $\psi_C$ , this means that the circuit  $C$  accepts the input tuple  $(a_1, \dots, a_l, b_1, \dots, b_l, c_1, \dots, c_l)$ . Thus,  $(i, j, k) \in S_C$ , where  $j = \#(b_1, \dots, b_l)$  and  $k = \#(c_1, \dots, c_l)$ .

We still need to show that  $j, k \in P$ . To see this, note that since  $M, T \models^\ell q_1 \cdots q_l \subseteq p_1 \cdots p_l$ , there exists  $w' \in T$  such that

$$w' \in V(p_t) \iff w \in V(q_t) \iff b_t = 1 \quad \text{for } 1 \leq t \leq l.$$

Thus,  $(b_1, \dots, b_l) \in T(p_1, \dots, p_n)$ , whence  $j \in P$ . Similarly we see that  $k \in P$ .

To prove the other implication, assume that  $C$  is a positive instance of the problem S-PER. Then there is an  $S_C$ -persistent set  $P \subseteq A_C$  such that  $2^l \in P$ . We let  $M = (W, R, V)$  be the Kripke model and  $T$  the team of  $M$  such that

- $T = W$  is the set of all tuples  $(a_1, \dots, a_m) \in \{0, 1\}^m$  that correspond to an accepting computation of  $C$  and for which  $\#(a_1, \dots, a_l), \#(a_{l+1}, \dots, a_{2l}), \#(a_{2l+1}, \dots, a_{3l}) \in P$ ,
- $R = \emptyset$ , and  $V(p_t) = \{(a_1, \dots, a_m) \in W \mid a_t = 1\}$  for  $1 \leq t \leq m$ .

We will now show that  $M, T \models^\ell \varphi_C$ , whence  $\varphi_C$  is satisfiable. Note first that  $M, T \models^\ell \psi_C$ , since by the definition of  $T$  and  $V$ , for any  $w \in T$ , the truth values of  $p_i$  in  $w$  correspond to an accepting computation of  $C$ .

To prove  $M, T \models^\ell q_1 \cdots q_l \subseteq p_1 \cdots p_l$ , assume that  $(b_1, \dots, b_l) \in T(q_1, \dots, q_l)$ . Then  $i := \#(b_1, \dots, b_l) \in P$ , and since  $P$  is  $S_C$ -persistent, there are  $j, k \in P$  such that  $(i, j, k) \in S_C$ . Thus, there is a tuple  $(a_1, \dots, a_m) \in \{0, 1\}^m$  corresponding to an accepting computation of  $C$  such that  $(a_1, \dots, a_l) = (b_1, \dots, b_l)$ ,  $j = \#(a_{l+1}, \dots, a_{2l})$  and  $k = \#(a_{2l+1}, \dots, a_{3l})$ . This means that  $(a_1, \dots, a_m)$  is in  $T$ , and hence  $(b_1, \dots, b_l) \in T(p_1, \dots, p_l)$ . The claim  $M, T \models^\ell r_1 \cdots r_l \subseteq p_1 \cdots p_l$  is proved in the same way.



Note that since  $M, T \models p_m$ , we have  $T(p_m, \dots, p_m) = \{(1, \dots, 1)\}$ . Furthermore, since  $2^l = \sharp(1, \dots, 1) \in P$  and  $P$  is  $S_C$ -persistent, there is an element  $(a_1, \dots, a_m) \in T$  such that  $(a_1, \dots, a_l) = (1, \dots, 1)$ . Thus, we see that  $(1, \dots, 1) \in T(p_1, \dots, p_l)$ , and consequently  $M, T \models^\ell p_m \cdots p_m \subseteq p_1 \cdots p_l$ .  $\square$

**Corollary 1.** *The satisfiability and finite satisfiability problems of modal inclusion logic with lax semantics are EXPTIME-complete with respect to PSPACE reductions.*

Note that the formula  $\varphi_C$  used in the proof of Theorem 2 is in *propositional inclusion logic*, i.e., it does not contain any modal operators. Thus, our proof shows that the satisfiability problem of propositional inclusion logic is already EXPTIME-hard. Naturally, this problem is also in EXPTIME, since propositional inclusion logic is a fragment of MInc.

**Corollary 2.** *The satisfiability and finite satisfiability problems of propositional inclusion logic with lax semantics are EXPTIME-complete with respect to PSPACE reductions.*

### 3.3 Upper bound for strict semantics

In this section we show that the satisfiability and finite satisfiability problems for MInc with strict semantics are in NEXPTIME. The proof is a simple adaptation of the upper bound argument for lax semantics, but uses *two-variable logic with counting*,  $\text{FOC}^2$ , which has NEXPTIME-complete satisfiability and finite satisfiability problems [10] (but no finite model property).

Let  $\theta$  be a formula of MInc. The equisatisfiable translation of  $\theta$  is obtained from the formula  $\varphi_\theta$ , which we defined when considering lax semantics. It is clear that  $\varphi_\theta$  translates via a simple extension of the *standard translation* into  $\text{FOC}^2$ ; see [1] for the standard translation of modal logic. Let  $t(\varphi_\theta)$  denote the  $\text{FOC}^2$ -formula obtained by using the (extension of the) standard translation. For each  $\varphi \in \text{SUB}(\varphi_\theta)$ , let  $t(\chi_\varphi)$  denote the translation of the subformula  $\chi_\varphi$  of  $\varphi_\theta$ ; see the argument for lax semantics for the definition of the formulas  $\chi_\varphi$ . The only thing we now need to do is to modify the formulas  $t(\chi_{\diamond\varphi})$  and  $t(\chi_{\varphi \vee \psi})$ .

In the case of  $t(\chi_{\varphi \vee \psi})$ , we simply add a conjunct stating that the unary predicates  $p_\varphi$  and  $p_\psi$  are interpreted as disjoint sets:  $\neg \exists x(p_\varphi(x) \wedge p_\psi(x))$ .

To modify the formulas  $t(\chi_{\diamond\varphi})$ , we appoint a novel binary relation  $R_{\diamond\varphi}$  for each formula  $\diamond\varphi \in \text{SUB}(\theta)$ . We then define the formula  $\beta$  which states that  $R_{\diamond\varphi}$  is a function from the interpretation of  $p_{\diamond\varphi}$  onto the interpretation of  $p_\varphi$ .

$$\begin{aligned} \beta := & \forall x(p_{\diamond\varphi}(x) \rightarrow \exists^1 y(R_{\diamond\varphi}xy \wedge p_\varphi(y)) \wedge \forall x \forall y(R_{\diamond\varphi}xy \rightarrow (p_{\diamond\varphi}(x) \wedge p_\varphi(y))) \\ & \wedge \forall y(p_\varphi(y) \rightarrow \exists x(p_{\diamond\varphi}(x) \wedge R_{\diamond\varphi}xy)). \end{aligned}$$

Define  $\beta' := \forall x \forall y(R_{\diamond\varphi}xy \rightarrow Rxy)$ , where  $R$  is the accessibility relation of modal inclusion logic. The conjunction  $\beta \wedge \beta'$  is the desired modification of  $t(\chi_{\diamond\varphi})$ .

The modification of  $t(\varphi_\theta)$ , using the modified versions of  $t(\chi_{\varphi \vee \psi})$  and  $t(\chi_{\diamond\varphi})$ , is the desired  $\text{FOC}^2$ -formula equisatisfiable with  $\theta$ . The proof of the following theorem is practically identical to the corresponding argument for lax semantics.

**Theorem 3.** *The satisfiability and finite satisfiability problems for MInc with strict semantics are in NEXPTIME.*

### 3.4 Lower bound for strict semantics

**Theorem 4.** *The satisfiability and finite satisfiability problems for MInc with strict semantics are NEXPTIME-hard.*

*Proof.* We will provide a chain of reductions from *Dependence-QBF-Validity* (in short DQBF-VAL) to *Inclusion-QBF-Validity* (in short IncQBF-VAL), and finally to satisfiability of MInc with strict semantics.

Peterson et al. [9] introduced a so-to-speak dependence version of QBF by extending the usual QBF syntax to allow stating on which universally quantified propositions an existentially quantified proposition solely depends. Instances of the problem are of the form  $(\forall p_1)(\exists q_1 \setminus P_1) \cdots (\forall p_k)(\exists q_k \setminus P_k) \varphi \quad (\star)$ , where each set  $P_i$  contains a subset of the propositions  $\{p_1, \dots, p_i\}$  quantified universally superordinate to  $q_i$ , and  $\varphi$  is a propositional logic formula in the variables  $\{p_1, \dots, p_k\} \cup \{q_1, \dots, q_k\}$ . The set  $P_i$  indicates that the choice for the value of  $q_i$  is given by a Boolean function that takes as inputs only the values of the variables in  $P_i$  (see [9] for the full details).

By well-known standard arguments in the field of team semantics, it is easy to show that the formula of Eqn.  $(\star)$  can be written in the alternative form (where  $\overline{p_i}$  lists the variables in  $P_i$ )

$$(\forall p_1)(\exists q_1) \cdots (\forall p_k)(\exists q_k) \left( \varphi \wedge \bigwedge_{i \in \{1, \dots, k\}} \text{dep}(\overline{p_i}, q_i) \right), \quad (1)$$

with the following semantics (where  $M$  is a Kripke model and  $T$  is a team).

- $M, T \models \forall p \psi$  iff  $M', T^p \models \psi$ , where  $T^p$  is obtained from  $T$  by simultaneously replacing each  $w \in T$  by two new worlds  $u, v$  that agree with  $w$  on all propositions other than  $p$ , and the points  $u, v$  disagree with each other on  $p$ .  $M'$  is obtained from  $M$  by modifying the domain  $W$  of  $M$  to the new domain  $W' = T^p \cup (W \setminus T)$ , and modifying the valuation of  $M$  to a new one that agrees with the specification of  $T^p$ ; outside  $T^p$  the new valuation agrees with the old one. The accessibility relation does not play a role here.
- $M, T \models \exists p \psi$  iff  $M', T_p \models \psi$ , where  $T_p$  is obtained from  $T$  by simultaneously replacing each  $w \in T$  by a new world  $u$  that agrees with  $w$  on propositions other than  $p$ , and may or may not agree with  $w$  on  $p$ . Similarly to the case above,  $M'$  is obtained from  $M$  by modifying the domain  $W$  of  $M$  to the new domain  $W' = T_p \cup (W \setminus T)$ , and modifying the valuation of  $M$  to a new one that agrees with the specification of  $T_p$ ; outside  $T_p$  the new valuation agrees with the old one. The accessibility relation does not play a role here.
- The connectives  $\vee$  and  $\wedge$  are interpreted exactly as in the case of modal inclusion logic using strict semantics. Literals  $p$ ,  $\neg p$  are also interpreted as in modal inclusion logic.

- $M, T \models \text{dep}(p_1, \dots, p_k, q)$  if each pair of worlds in  $T$  that agree on the truth values of each of the propositions  $p_1, \dots, p_k$ , also agree on the value of  $q$ .

Our formulation of the DQBF-VAL problem of Peterson et al. [9], with alternative inputs such as those in Eqn. 1, is equivalent to the original problem. Peterson et al. showed that their problem lifts the computational complexity from PSPACE-completeness (for the standard quantified Boolean formula validity) to in fact NEXPTIME-completeness.

Inclusion-QBF (IncQBF) is a language obtained from our formulation of the Dependence-QBF (DQBF). It translates the expressions  $\text{dep}(p_1, \dots, p_k, q)$  to inclusion atoms in the way we next describe. Inspired by Galliani et al. [3], we observe that inclusion atoms can simulate formulas  $\text{dep}(p_1, \dots, p_k, q)$ , as the following example demonstrates:  $\forall p \forall q \exists r (\text{dep}(q, r) \wedge \varphi)$  is equivalent to  $\forall p \forall q \exists r (\forall s (sqr \subseteq pqr) \wedge \varphi)$ , where  $\varphi$  is a formula of propositional logic. This can be generalized to work for expressions with conjunctions of atoms  $\text{dep}(p_1, \dots, p_k, q)$  for arbitrary  $k$ .

Now, for the last step, we need to explain how IncQBF-VAL finally reduces to MInc-strict-SAT. This is just a slight modification of the standard proof of Ladner showing PSPACE-hardness of plain modal logic via a reduction from QBF validity [7]. The idea is to enforce a complete assignment tree. Further, one uses clause propositions which are true if the corresponding literal holds. Let us denote the formula which enforces the described substructure by  $\varphi_{\text{struc}}$  (for details, see [7]). The final formula is obtained from an IncQBF-VAL instance  $\exists r_1 \forall r_2 \dots \exists r_n (\varphi \wedge \chi)$  where  $\varphi$  is the conjunctive normal form formula and  $\chi$  is the conjunction of the inclusion atoms (stemming from the translation above); the final formula is then a formula of type  $\varphi_{\text{struc}} \wedge \Diamond \Box \dots \Delta (\varphi \wedge \chi)$ , where  $\Delta = \Box$  if  $\exists = \forall$  and  $\Delta = \Diamond$  if  $\exists = \exists$ . Let us denote this translation by the function  $f$  which can be computed in polynomial time. Then it is easy to verify that  $\varphi \in \text{IncQBF-VAL}$  iff  $f(\varphi) \in \text{MInc-strict-SAT}$ . It is straightforward to observe that this covers also the case for finite satisfiability.  $\square$

**Corollary 3.** *The satisfiability and finite satisfiability problems of modal inclusion logic with strict semantics are NEXPTIME-complete.*

## 4 Conclusion

We have compared the strict and lax variants of team semantics from the perspective of satisfiability problems for modal inclusion logic MInc. Interestingly, the problems differ in complexity. Strict semantics leads to NEXPTIME-completeness, while lax semantics gives completeness for EXPTIME. For the journal version we plan to include a stronger polynomial-time reduction result for the EXPTIME lower bound of MInc-lax-SAT. In the future it will be interesting to study model checking problems for MInc under strict and lax semantics. Also, the complexity of validity problems for MInc and, related to this, proof-theoretic properties of the logic remain to be investigated.

**Acknowledgements.** The authors thank the anonymous referees for their comments. The third author is supported by DFG grant ME 4279/1-1. The second author acknowledges support from Jenny and Antti Wihuri Foundation.

## References

- [1] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, 2001.
- [2] P. Galliani. Inclusion and exclusion dependencies in team semantics - on some logics of imperfect information. *Ann. Pure Appl. Logic*, 163(1):68–84, 2012.
- [3] P. Galliani, M. Hannula, and J. Kontinen. Hierarchies in independence logic. In S. Ronchi Della Rocca, editor, *Proceedings of Computer Science Logic 2013, Leibniz International Proceedings in Informatics (LIPIcs)*, volume 23, pages 263–280, 2013.
- [4] R. Greenlaw, H. J. Hoover, and W. L. Ruzzo. *Limits to Parallel Computation: P-Completeness Theory*. Oxford University Press, 1995.
- [5] E. Hemaspaandra. The price of universality. *Notre Dame Journal of Formal Logic*, 37(2):174–203, 1996.
- [6] J. Kontinen, J.-S. Müller, H. Schnoor, and H. Vollmer. Modal independence logic. In R. Goré, B. P. Kooi, and A. Kurucz, editors, *Proc. of Advances in Modal Logic*, volume 10, pages 353–372. College Publications, 2014.
- [7] R. Ladner. The computational complexity of provability in systems of modal propositional logic. *SIAM Journal on Computing*, 6(3):467–480, 1977.
- [8] C. H. Papadimitriou. *Computational complexity*. Addison-Wesley, 1994.
- [9] G. Peterson, S. Azhar, and J. H. Reif. Lower bounds for multiplayer non-cooperative games of incomplete information. *Computers and Mathematics with Applications*, 41:957–992, 2001.
- [10] I. Pratt-Hartmann. Complexity of the two-variable fragment with counting quantifiers. *JoLLI*, 14(3):369–395, 2005.
- [11] M. Sevenster. Model-theoretic and computational properties of modal dependence logic. *Journal of Logic and Computation*, 19(6):1157–1173, 2009.
- [12] J. Väänänen. *Dependence Logic*. Cambridge University Press, 2007.
- [13] J. Väänänen. Modal Dependence Logic. In K. Apt and R. van Rooij, editors, *New Perspectives on Games and Interaction*, pages 237–254. Amsterdam University Press, 2008.
- [14] J. van Eijck. Dynamic epistemic logics. In A. Baltag and S. Smets, editors, *Johan van Benthem on Logic and Information Dynamics*, pages 175–202. Springer, 2014.

## Appendix

Let  $M = (\Sigma, Q, \gamma, s_0, \delta)$  be an alternating Turing machine. Thus,  $\Sigma$  is a finite tape alphabet,  $Q$  is a finite set of states, the function  $\gamma : Q \rightarrow \{\forall, \exists, Acc, Rej\}$  divides  $Q$  according to the type of the states (universal, existential, accepting, rejecting),  $s_0 \in Q$  is the initial state, and  $\delta : \Sigma \times Q \rightarrow \mathcal{P}(\Sigma \times Q \times \{left, right, 0\})$  is a transition function.

Configurations of  $M$  are defined as usual. If  $\alpha$  is a configuration, we write  $s^\alpha$  for its state. Furthermore, we write  $\alpha \mapsto_M \beta$  if  $\alpha$  and  $\beta$  are configurations such that  $\beta$  can be obtained from  $\alpha$  by a transition allowed by  $\delta$ . Without loss of generality we assume that  $\Sigma = \{0, 1\}$ , and  $|\delta(0, s)| = |\delta(1, s)| = 2$  for all  $s$  such that  $\gamma(s) \in \{\forall, \exists\}$ . Thus, if  $\gamma(s^\alpha) \in \{\forall, \exists\}$  for a configuration  $\alpha$ , then there are exactly two configurations  $\beta$  such that  $\alpha \mapsto_M \beta$ . On the other hand, if  $\gamma(s) \in \{Acc, Rej\}$ , we assume that  $\delta(0, s) = \delta(1, s) = \emptyset$ . Thus, the computation halts in a configuration  $\alpha$  such that  $\gamma(s^\alpha) \in \{Acc, Rej\}$ .

The sets  $AC(M)$  of accepting configurations and  $RC(M)$  of rejecting configurations of  $M$  are defined recursively in the usual way:

- If  $\gamma(s^\alpha) = Acc$ , then  $\alpha \in AC(M)$ .
- If  $\gamma(s^\alpha) = \forall$  and  $\beta \in AC(M)$  for all  $\beta$  such that  $\alpha \mapsto_M \beta$ , then  $\alpha \in AC(M)$ .
- If  $\gamma(s^\alpha) = \exists$  and there is  $\beta \in AC(M)$  such that  $\alpha \mapsto_M \beta$ , then  $\alpha \in AC(M)$ .
- If  $\gamma(s^\alpha) = Rej$ , then  $\alpha \in RC(M)$ .
- If  $\gamma(s^\alpha) = \forall$  and there is  $\beta \in RC(M)$  such that  $\alpha \mapsto_M \beta$ , then  $\alpha \in RC(M)$ .
- If  $\gamma(s^\alpha) = \exists$  and  $\beta \in RC(M)$  for all  $\beta$  such that  $\alpha \mapsto_M \beta$ , then  $\alpha \in RC(M)$ .

The machine  $M$  accepts (rejects) a word  $w \in \Sigma^*$  if  $\alpha_w \in AC(M)$  ( $\alpha_w \in RC(M)$ , respectively) for the initial configuration  $\alpha_w$  of  $M$  with  $w$  as input. We denote the language  $\{w \in \Sigma^* \mid M \text{ accepts } w\}$  by  $L_M$ . The machine  $M$  decides the language  $L_M$ , if in addition  $M$  rejects all inputs  $w \notin L_M$ .

The class **APSPACE** consists of all languages  $L_M$ , where  $M$  is an alternating Turing machine that uses only polynomial number of tape cells. It is well known that if  $L \in \text{APSPACE}$ , then there is a polynomial space alternating machine  $M$  that decides  $L$ , and which is acyclic in the sense that there are no  $\mapsto_M$ -cycles among the configurations of  $M$ .

*Proof of Proposition 1:* Let  $L \in \text{APSPACE}$ , and let  $M$  be an alternating Turing machine that works in polynomial space such that  $L = L_M$ . For each input word  $w \in \{0, 1\}^*$  we construct a circuit  $C_{M,w}$  in polynomial time from  $w$  such that  $C_{M,w}$  is a positive instance of S-PER if and only if  $M$  accepts  $w$ . This shows that  $L_M$  is reducible to S-PER, and since this holds for every language  $L_M$  in **APSPACE**, and **APSPACE** = **EXPTIME**, it follows that S-PER is **EXPTIME**-hard.

As explained above, we may assume that  $\mapsto_M$  is acyclic, and  $M$  decides the language  $L$ . Let  $f$  be the polynomial such that for all inputs of length  $n$ ,  $M$  uses at most  $f(n)$  tape cells. Thus, if  $w = w_1 \dots w_n \in \{0, 1\}^n$  is an input word for  $M$ , then we can encode the possible configurations of  $M$  during the computation on input  $w$  with tuples

$$(a_1, \dots, a_{2m+k}) \in \{0, 1\}^{2m+k},$$

where  $m := f(n)$ , as follows:

- $(a_1, \dots, a_m)$  represents the contents of the tape in  $\alpha$ ,
- $(a_{m+1}, \dots, a_{2m})$  encodes the position of the tape head in  $\alpha$ :  $a_{m+i} = 1$  iff the head is on the  $i$ -th cell,
- $(a_{2m+1}, \dots, a_{2m+k})$  encodes the state  $s^\alpha$ :  $a_{2m+(i+1)} = 1$  iff  $s^\alpha = s_i$ , where  $s_0, \dots, s_{k-1}$ , lists  $Q$  in some fixed order,

The circuit  $C_{M,w}$  will now be defined in such a way that the following conditions hold:

1.  $C_{M,w}$  has  $3l$  input gates, where  $l = 2m + k$ .
2. If  $\vec{a} = (a_1, \dots, a_l) \in \{0, 1\}^l$  is a tuple which encodes a configuration  $\alpha$  such that  $\gamma(s^\alpha) = Acc$ , then  $C_{M,w}$  accepts the input  $\vec{a} \vec{a} \vec{a}$ .
3. If  $\vec{a} = (a_1, \dots, a_l)$ ,  $\vec{b} = (b_1, \dots, b_l)$  and  $\vec{c} = (c_1, \dots, c_l)$  are tuples in  $\{0, 1\}^l$  which encode configurations  $\alpha$ ,  $\beta_1$  and  $\beta_2$  such that  $\beta_1 \neq \beta_2$ ,  $\gamma(s^\alpha) = \forall$ ,  $\alpha \mapsto_M \beta_1$  and  $\alpha \mapsto_M \beta_2$ , then  $C_{M,w}$  accepts the input  $\vec{a} \vec{b} \vec{c}$ .
4. If  $\vec{a} = (a_1, \dots, a_l)$  and  $\vec{b} = (b_1, \dots, b_l)$  are tuples in  $\{0, 1\}^l$  which encode configurations  $\alpha$  and  $\beta$  such that  $\gamma(s^\alpha) = \exists$  and  $\alpha \mapsto_M \beta$ , then  $C_{M,w}$  accepts the input  $\vec{a} \vec{b} \vec{b}$ .
5. If  $\vec{a} = (1, \dots, 1) \in \{0, 1\}^l$  and  $\vec{b} = (b_1, \dots, b_l) \in \{0, 1\}^l$  is a tuple that encodes the initial configuration  $\alpha_w$  of  $M$  with input word  $w$ , then  $C_{M,w}$  accepts the input  $\vec{a} \vec{b} \vec{b}$ .
6.  $C_{M,w}$  does not accept any other input tuples  $(a_1, \dots, a_{3l}) \in \{0, 1\}^{3l}$ .

Clearly the conditions 1-6 above can be checked in polynomial time with respect to  $l$ , and hence with respect to the length  $n$  of the input  $w$ . Thus, the circuit  $C_{M,w}$  can be constructed in polynomial time from the input word  $w$ .

Assume first that  $M$  accepts the input  $w$ . Then the initial configuration  $\alpha_w$  of  $M$  with input  $w$  is in the set  $AC(M)$ . Consider now the structure  $\mathfrak{A}_C = (A_C, S_C)$  defined by the circuit  $C := C_{M,w}$ . Let  $P_0 \subseteq A_C$  be the set of all  $i = \sharp(a_1, \dots, a_l)$  such that  $(a_1, \dots, a_l)$  encodes a configuration  $\alpha \in AC(M)$ . Using conditions 2-4 and the definition of  $AC(M)$  it is easy to show that  $P_0$  is  $S_C$ -persistent. But then, by condition 5,  $P = P_0 \cup \{\sharp(1, \dots, 1)\}$  is an  $S_C$ -persistent set such that  $2^l \in P$ , whence  $C$  is a positive instance of S-PER.

Assume then that  $C := C_{M,w}$  is a positive instance of S-PER. Then there is an  $S_C$ -persistent set  $P$  such that  $2^l = \sharp(1, \dots, 1) \in P$ . Let  $P^M$  be the set of all configurations  $\alpha$  of  $M$  such that  $\sharp(a_1, \dots, a_l) \in P$  for the tuple  $(a_1, \dots, a_l)$  that encodes  $\alpha$ . By conditions 5 and 6, the initial configuration  $\alpha_w$  is in  $P^M$ . Thus, it suffices to show that  $P^M \subseteq AC(M)$ .

Suppose this is not the case, i.e.,  $P^M \setminus AC(M) \neq \emptyset$ . Since  $P^M$  is finite, and  $\mapsto_M$  is acyclic, then there exists a configuration  $\alpha \in P^M \setminus AC(M)$  which does not have  $\mapsto_M$ -successors in  $P^M \setminus AC(M)$ . We divide the argument into cases according to the type  $\gamma(s^\alpha)$  of the state of  $\alpha$ .

- Observe first that  $\gamma(s^\alpha) = Acc$  is not possible, since  $\alpha \notin AC(M)$ .

- Assume that  $\gamma(s^\alpha) = \text{Rej}$ . Let  $(a_1, \dots, a_l) \in \{0, 1\}^l$  be the tuple that encodes  $\alpha$ . Then by conditions 2-6, there are no tuples  $(b_1, \dots, b_l), (c_1, \dots, c_l)$  such that  $(\#(a_1, \dots, a_l), \#(b_1, \dots, b_l), \#(c_1, \dots, c_l)) \in S_C$ . This means that  $\alpha \notin P^M$ , contrary to our assumption.
- If  $\gamma(s^\alpha) = \forall$ , then by conditions 3 and 6 we see that  $\beta_1, \beta_2 \in P^M$ , where  $\beta_1$  and  $\beta_2$  are the  $\mapsto_M$ -successors of  $\alpha$ . Since  $\alpha$  has no  $\mapsto_M$ -successors in  $P^M \setminus \text{AC}(M)$ , we have  $\beta_1, \beta_2 \in \text{AC}(M)$ . But then by the definition of  $\text{AC}(M)$ , also  $\alpha \in \text{AC}(M)$ , contrary to our assumption.
- If  $\gamma(s^\alpha) = \exists$ , then by conditions 4 and 6,  $\beta_1 \in P^M$  or  $\beta_2 \in P^M$ , where  $\beta_1$  and  $\beta_2$  are the  $\mapsto_M$ -successors of  $\alpha$ . Since  $\alpha$  has no  $\mapsto_M$ -successors in  $P^M \setminus \text{AC}(M)$ , it follows that either  $\beta_1 \in \text{AC}(M)$  or  $\beta_2 \in \text{AC}(M)$ . Hence, by the definition of  $\text{AC}(M)$ , we have  $\alpha \in \text{AC}(M)$ , contrary to our assumption.

Since all the cases lead to contradiction, we conclude that  $P^M \subseteq \text{AC}(M)$ .  $\square$